

Домашнее задание 4

Секундомер

На этой неделе потребуется разработать простой цифровой секундомер, отслеживающий время с точностью до одной десятой доли секунды. Секундомер должен управляться кнопками «Старт», «Стоп» и «Сброс». Для создания секундомера потребуется совместить работу с таймером, форматирование строк и вывод текста на холст. Для упрощения работы рекомендуется использовать шаблон `stopwatch_template.py`.

Рекомендованный порядок работы над заданием

1. Создайте таймер с интервалом срабатывания 0.1 секунды, обработчик которого будет увеличивать целое число, хранящееся в глобальной переменной. Это число будет отслеживать время. Не забывайте, что функции `create_timer()` время задаётся в миллисекундах (1мс – одна тысячная секунды).
Проверьте, что таймер работает правильно, печатая значение переменной времени в консоли.
Не используйте для хранения времени числа с плавающей точкой. Скорее всего, такой вариант можно будет заставить работать, но только с приложением больших усилий на борьбу с погрешностями. Храните время с помощью целого числа, например $12 = 1.2$ секунды.
2. Напишите обработчик рисования, который отобразит время в виде числа в центре холста. Пока не задумывайтесь о красивом форматировании, просто используйте `str()` чтобы преобразовать число в строку и вывести как есть.
3. Добавьте кнопки «Старт» и «Стоп», обработчики которых будут запускать и останавливать таймер.
Затем добавьте кнопку «Сброс», которая останавливает таймер и сбрасывает время в 0. При запуске программы секундомер должен быть остановлен.
4. Напишите вспомогательную функцию `format(t)`, которая должна возвращать строку в формате A:BC.D, где A – минуты, BC – секунды (B – цифра от 0 до 5), D – десятые доли секунд.
Проверьте функцию `format(t)` с помощью `doctest`'ов (тестов в строке документации). Обратите внимание, чтобы возвращаемая функцией строка всегда содержала необходимые ведущие нули:
 - `format(0) = 0:00.0`
 - `format(11) = 0:01.1`
 - `format(321) = 0:32.1`
 - `format(613) = 1:01.3`Подсказка: используйте целочисленное деление (`//`) и остаток от деления (`%`) чтобы получить значения A, B, C и D из исходного числа.
5. Добавьте вызов функции `format()` в обработчик рисования, чтобы получить законченный секундомер.

Дополнительные расширения

1. Превратите секундомер в тренажер реакции. Для этого добавьте к секундомеру два счётчика, которые будут отслеживать, сколько раз секундомер был остановлен, и сколько из них его удалось остановить в ровную секунду (1.0, 2.0, 3.0 и т.д.). Значение этих

- счётчиков должно отображаться в верхней правой части холста в формате “x/y”, где y – общее число остановок секундомера, x – число остановок в ровную секунду.
2. Добавьте код, который не позволит увеличивать счётчики нажатием на кнопку «Стоп», если секундомер уже остановлен.
Подсказка: для этого может быть полезно добавить булеву переменную, которая будет иметь значение True, если секундомер запущен, и False – если остановлен.
 3. Модифицируйте обработчик кнопки «Сброс» так, чтобы он сбрасывал значения счётчиков в 0 при сбросе секундомера.

Оценка задания

Полностью работоспособный секундомер, с функциональностью, описанной в разделе «Рекомендованный порядок выполнения задания»

- 70%;

Форматирование времени реализовано некорректно

- -30%;

Форматирование времени реализовано корректно, но отсутствуют doctest’ы

- -15%;

Добавлены счётчики нажатий из «дополнительных расширений»

- 15%;

Кнопки «Стоп» и «Сброс» корректно работают с подсчётом нажатий

- 15%.

Срок сдачи задания:

для подгруппы, занимающейся по вторникам: 11 октября 2016;

для подгруппы, занимающейся по пятницам: 14 октября 2016.

В случае сдачи задания с опозданием, полученные за него баллы будут уменьшены:

при задержке на 1 неделю: баллы умножаются на 0.9;

при задержке на 2 недели: баллы умножаются на 0.75;

при задержке на 3 и более недель: баллы умножаются на 0.65.